Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L2	51	servlet$4 near6 (translat$4 or conver$6) and "709"/$.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/09/15 16:52

TDB-ACC-NO: NNRD422124

DISCLOSURE TITLE: A Process for Selective Routing of Servlet Content to Transcoding Modules

PUBLICATION-DATA: Research Disclosure, June 1999, UK

VOLUME NUMBER: 42

ISSUE NUMBER: 422

PUBLICATION-DATE: June 1, 1999 (19990601)

CROSS REFERENCE: 0374-4353-42-422-0

DISCLOSURE TEXT:

The use of servlets is becoming an increasingly popular approach for generating web content. Also becoming increasingly prevalent is the use of mobile and wireless handheld pervasive computing (PvC) devices to access Web content such as that generated by servlets. These type of devices have browsing and screen size capabilities that are drastically different from desktop workstations, and because of this, require content to be custom tailored (i.e., transcoded) for the pervasive computing device. For example, custom tailoring may include the removing or shrinking of images, the creation of summary pages of headings, page splitting, etc.

Thus, one important issue that must be addressed in order to be able to support PvC devices is the ability to route servlet generated content to some notion of a transcoding engine that is able to tailor or reformat the content for the PvC device on which it is intended to be displayed. Simply routing all content generated by servlets to a transcoding engine is not an acceptable solution as it can lead to unnecessary performance degradation for the common case when transcoding is not required . This performance degradation is due to the fact that transcoding engines typically employ some complex notion of rules that orchestrate the invocation of one or more

transcoding modules. While the overhead for this process is negligible when compared to the overhead of actually performing most types of transcoding, this overhead is not acceptable for servlet generated content for which it can be determined that no transcoding is required.

In this disclosure, we present a process that permits servlet generated content to be selectively routed to a transcoding module in an efficient manner by quickly recognizing content that is a candidate for needing transcoding support. Thus, servlet generated content that is clearly not a candidate for transcoding is dynamically routed such that it avoids going through the transcoding engine and instead directly sent to the user's browser. As a result, the extra overhead that results from the use of the transcoding rules engine is avoided. Servlet generated content that is identified as requiring transcoding is selectively routed to the transcoding engine, permitting the necessary transcoding to be performed, and of course incurring the overheads associated with this process.

The solution is a process whereby user and device preferences, which can be queried in a highly efficient manner, are the basis for characterizing the expected servlet generated content as achieving candidacy for being sent to the transcoding engine. For example, a user in a particular session may have the following device preferences:

user=uniqueUserAndSessionId UserAgentWinCE=1
UserAgentThinkPad=0 UserAgentPalmPilot=1

Thus, for this particular user, the content generated by the servlet should be sent to the transcoder whenever the user is accessing the web server from a WinCE device or a Palm Pilot device.

When the user is accessing the web server from a ThinkPad (or other desktop machine), the servlet generated content is recognized as not being a candidate for transcoding and thus forwarded directly to

the
device for rendering.

One difficult issue that remains is the ability to associate the above preferences with a user for a particular session of HTTP requests. To accomplish this, cookies are utilized to manage state information. To facilitate efficient and expedient processing of preference dependent requests, the transmission of cookies is minimized and preferences are cached by the **servlet** invocation system by associating them with the users current IP address. The user preferences originate from a preference configuration page and device preferences are associated with the User-Agent field of the HTTP Header such that the User-Agent value represents the persistent and transient device preference key.

Upon processing of the preference configuration page, user preferences are stored in a persistent repository, and a set-cookie header with a unique path, null domain, security specified, and an excessively large expiration period is generated. The cookie name is set to "persistent_session_id" with a unique session identifier, which is generated by the **servlet** invocation system:

- PersistentSessionIdentifier = f(serverIPAddress,currentTime, threadID)

It is worth noting that the User Agent IP Address does not suffice as the persistent session key, because in many cases IP Addresses are acquired dynamically through DHCP protocols. The User Agent IP Address is utilized as a "transient session identifier" however.

The configured property set is cached in a volatile table and indexed by the IP address of the user agent. The property set is stored in a persistent repository, which is indexed by the unique session identifier encapsulated by the session cookie. As HTTP requests are processed by the **servlet** engine, user preferences are accessed by cache hit's indexed by the user agent IP address. If no such session cookie exists, the system may opt to serve up a preference configuration page to acquire initial preference settings.

Upon resolution of the persistent session identifier, the persisted preferences are loaded and stored in the volatile cache, indexed by the User-Agent IP Address. After resolution of the persistent session identifier another HTTP redirect is invoked so as to request the original page, which generated the cache miss. Due to the minimization of cookie transmissions, the cached user preferences are replaced via a time-out scheme to avoid potential confusion with stale IP addresses in DHCP networks.

If the resolved user preference set implies image transcoding is desired, the servlet forwarding algorithm invokes the registered image transcoding servlet. A cache miss implies the persistent session key must be resolved. This is currently achieved via an HTTP redirect, such that the user agent requests a nonexistent page from the host server. Such a request satisfies the session cookies path condition, such that the session cookie is transmitted to the host server and the session identifier is resolved. Security is maintained via the Http cookie "secure" attribute, which informs the user agent to transmit the cookie in a secure fashion.

Another way to mitigate this difficulty is to leverage session support that may be provided by the servlet engine itself. For example, WebSphere provides a mechanism for creating sessions where by a servlet can request a session object and can use this session object to cache the user preferences that need to be associated with the session. In summary, the selective routing of servlet content to transcoding modules is based on a process for determining candidacy for this functionality that relies upon low overhead lookup of user and device preferences in the context of a session. The system is efficient in that protocol flows are minimized and preferences are cached in a volatile table.

Servlet generated content that is clearly not a candidate for transcoding is dynamically routed such that it avoids going through the transcoding engine and instead directly sent to the user's browser. As a result, the extra overhead that results from the use of the transcoding rules engine is avoided. Servlet

generated content that is identified as requiring transcoding can be dynamically routed to the transcoding engine, thus permitting the servlet to take full advantage of the content tailoring features provided by transcoding engine.

There are two other known approaches to this problem.

First, all servlet generated content can be sent to some form of transcoding engine such as an intermediary transforming proxy. As discussed above, this solution results in sending all content to a transcoding engine and results in all servlet generated content incurring overhead from the transcoding engine. In the second approach, each servlet is responsible for being able to transcode the content it generates to a variety of formats in order to permit its content to be delivered to a variety of pervasive computing devices. The drawback of this approach is that whenever it becomes necessary to support a new transcoding capability, all servlets that exist on the system that generate content have to be modified such that they support the new transcoding capability.

US-PAT-NO: 5928323

DOCUMENT-IDENTIFIER: US 5928323 A
See image for Certificate of Correction

TITLE: Apparatus and method for dynamically generating information with server-side software objects

---------- KWIC ---------

Detailed Description Text - DETX (12):
After the thread pool administration operations are performed (step 74) a thread retrieves a request from the connection queue (step 76). The thread then maps the request to a servlet name (step 78). The servlet may be specified by a URL, in which case the mapping process is direct. On the other hand, some translation process may be required to identify which servlet will be able to service the request. The mapping operation may be performed in one of the following ways. A server administrator may specify that some kinds of client requests always map to a particular servlet. For example, one which talks to a particular database. A server administrator may specify that part of the client request is the name of the servlet, as found in an administered servlets directory. At many sites, that directory would be shared between servers which share the load of processing for the site's clients. Some servers may be able to automatically invoke servlets to filter the output of other servlets, based on their administrative configuration. For example, particular types of servlet output may trigger post-processing by other servlets, perhaps to perform format conversions. Properly authorized clients may specify the servlet to be invoked, without administrative intervention.

Current US Original Classification - CCOR (1):

709/203

Current US Cross Reference Classification - CCXR (1):
709/217

Current US Cross Reference Classification - CCXR (2):
709/219

US-PAT-NO: 6678518

DOCUMENT-IDENTIFIER: US 6678518 B2
See image for Certificate of Correction

TITLE: Dynamic content filter in a gateway

---------- KWIC ---------

Brief Summary Text - BSTX (11):
An advantage of the present invention is that a gateway server may be adapted to convert new content types and use scenarios without requiring new hardware implementations. The advantage is realized by configuring servlets (written, for example, in Java.TM.) to perform content conversions.

Brief Summary Text - BSTX (12):
According to an aspect of the invention, the method uses Java.TM. servlets in a gateway server to process resource requests from a mobile station and requested resources from an origin server in a wide area network. The servlets are arranged in a plurality of chains such that output from one servlet is processed by a subsequent servlet in the same chain. The servlets communicate with an administrator module, for invoking appropriate servlets, in the gateway server through an application programming interface. The servlets are configured to perform content conversions so as to adapt the requested content in accordance with user preferences, to optimize the content to a user device, to perform graphics conversions, or to automatically translate from one language to another, etc.

Detailed Description Text - DETX (6):
The administrator module 20 preferably identifies a portion of the mobile station's URL request (e.g., the protocol and the server address) and invokes one or more servlets 22, 24 (which may be arranged as a servlet chain) suitable

for processing that particular URL request based on the identified portion of the URL request. In other words, the administrator module 2015 maps the mobile station's URL request to a particular one or more servlets in accordance with predetermined parameters in the mobile station's URL request. The servlets 22, 24 process the request by for example formatting the WML encoded URL request into HTTP format and fetching the requested resource (e.g., a document formatted in HyperText Markup Language (HTML)) from the specified origin or web server. Upon receipt of the requested resource, the servlets 22, 24 preferably indicate to the administrator module 20 in Multipurpose Internet Mail Extensions (MIME) format the type of content (e.g., text, image, audio, video, message, etc.) received from the origin server 18. The administrator module 20 then invokes the appropriate servlet(s) 26 (or filters), based on the indicated content type, to further process the resource or content. Advantageously, the servlets 26 are configured to perform content conversions such as translating text from one language to another language (e.g., English to Finnish, or one computer language to another), or converting graphical data from one format to another (e.g., from Graphics Interchange Format (GIF) to Joint Photographic Experts Group (JPEG) format). The converted content is then translated into the Wireless Markup Language so that it may be properly processed by the user agent 15 of the requesting mobile station.

Detailed Description Text - DETX (7):

As shown in FIG. 2, the mobile station 12, through its user agent 15, transmits a URL request through the wireless network elements (e.g., the base station 14 and the switching control point 16, which are not shown in this drawing in order to avoid unnecessary distraction from the underlying invention) to the gateway server 10. The administrator module 20 receives the request and identifies one or more servlets 22, 24 (or servlet

chains) for
processing the request based on the characteristics or parameters of the URL
request. The identified servlets 22, 24 then execute the request and send it
to the origin server 18 using HTTP or other suitable formats or protocols. The
origin server 18 receiving the request fetches the requested resource and
returns the requested resource back to the requesting servlet 24. The
requesting servlet 24 then sends a response including a MIME header containing
information indicating the content type of the requested resource to the
administrator module 20. The administrator module 20 determines whether
further processing by other servlets is needed. If so, appropriate servlets
are invoked to convert the indicated content type to another content type,
desired or required by the user. If not, the requested resource is formatted
into Wireless Markup Language, using a filter or an assigned servlet, for
transmission to the mobile station 12.

Claims Text - CLTX (1):
1. A method for dynamically converting data between a mobile station in a
wireless communication network and a resource server in a wide area network
(WAN), the method comprising in sequence the steps of: (a) sending from the
mobile station to a gateway server in the wireless communication network a
request for a resource located on the resource server; (b) identifying, by an
administrative module in the ateway server, a request chain of Java servlets
appropriate for processing the request based on parameters contained in the
request; (c) selectively invoking, by the administrative module, the request
Java servlet chain identified in step (b); (d) re-formatting, by at least one
Java request conversion servlet in the request Java servlet chain, the request
for transmission over the WAN to the resource server; (e) selectively

invoking, by the at least one Java request conversion servlet, a next Java servlet in the request Java servlet chain; (f) sending, by at least one Java fetch servlet in the request Java servlet chain, the re-formatted request from the gateway server over the WAN to the resource server; (g) receiving, by the at least one Java fetch servlet, the requested resource from the resource server at the gateway server; (i) selectively invoking, by the at least one Java fetch servlet, at least one Java content-parsing servlet; (j) sending, by the at least one Java fetch servlet, the requested resource to the at least one Java content-parsing servlet; (k) generating, by the at least one Java content-parsing servlet, information indicating a content type of the received requested resource; (l) selectively invoking at least one Java content conversion servlet appropriate for the content type of the received resource; and (m) converting, by the at least one Java contend conversion, servlet, the content type of the received requested resource to a different content type.

Claims Text - CLTX (2):
2. The method of claim 1, wherein the at least one Java content conversion servlet comprises a plurality of Java servlets sequentially linked to each other as a content conversion chain of Java servlets.

Claims Text - CLTX (5):
5. The method of claim 1, wherein step (l) comprises the step of: selectively invoking the at least one Java content conversion servlet by one of the at least one Java content-parsing servlet and the at least one Java fetch servlet, based on the information indicating content type of the received requested resource.

Claims Text - CLTX (8):
8. The method of claim 1, further comprising the step of: instantiating, by the gateway server, at least one of the at least one Java request

conversion
servlet, the at least one Java fetch servlet, the at least one Java content-parsing servlet, and the at least one Java content conversion servlet
upon startup of the gateway server.

Claims Text - CLTX (10):
10. A system for dynamically converting data between a mobile station in a
wireless communication network and a resource server in a wide area network
(WAN), the system comprising: a user agent running in the mobile station for
sending a request for a resource located on the resource server; an administrator module in a gateway server for identifying and invoking a request
chain of Java servlets appropriate for processing the request; the request
chain of Java servlets comprising: at least one Java request conversion servlet
for re-formating the request for transmission over the WAN to the resource
server, and for invoking a next request Java servlet in the request Java
servlet chain; at least one Java fetch servlet for sending the re-formatted
request to the resource server, for receiving the requested resource from the
resource server, and for selectively invoking at least one Java content-parsing
servlet; said at least one Java content-parsing servlet for processing the
received requested resource and for generating information indicating content
type of the requested resource; and at least one Java content conversion
servlet for converting the content type of the received requested resource to a
different content type.

Claims Text - CLTX (12):
12. The system of claim 10, wherein the at least one Java content conversion servlet comprises a plurality of Java servlets sequentially linked
to each other as a content conversion chain of Java servlets.

Claims Text - CLTX (14):
14. The system of claim 10, wherein the request Java servlet chain, the at
least one Java content-parsing servlet, and the at least one Java

content
conversion servlet are linked together to form a larger chain of Java servlets,
wherein each servlet (except the last servlet) in the chain selectively invokes
a subsequent servlet in the chain.

Claims Text - CLTX (16):
16. A gateway server for dynamically converting data transmitted between a
mobile station in a wireless communication network and a resource server in a
wide area network, the system comprising: an administrator module for receiving
a request from the mobile station for a resource located on the resource
server; a Java environment for processing and transmitting the request to the
resource server, for receiving the requested resource from the resource server,
and for processing and transmitting the resource to the mobile station, said
Java environment comprising: a request chain of Java servlets comprising: at
least one Java request conversion servlet for re-formatting the request for
transmission over the WAN to the resource server, and for invoking a next
request Java servlet in the request Java servlet chain; at least one fetch
Java servlet for sending the re-formatted request to the resource server, for
receiving the requested resource from the resource server, and for selectively
invoking at least one content-parsing Java servlet; said at least one Java
content-parsing servlet for processing the received requested resource and for
generating information indicating content typ of the requested resource; and
at least one Java content conversion Java servlet for converting tee content
type of the received requested resource to a different content type.

Claims Text - CLTX (19):
19. The gateway server of claim 16, wherein the request Java servlet chain,
the at least one Java content-parsing servlet, and the at least one Java
content-conversion servlet are linked to each other to form a larger

chain of
Java servlets, wherein each servlet (except the last servlet) in the chain
selectively invokes a subsequent servlet in the chain.

Current US Cross Reference Classification - CCXR (7):
709/227